Übungen zum Kapitel 9

Plotten und Visualisieren

Erstellt und überarbeitet: armin.baenziger@zhaw.ch, 2. März 2020

In [1]:
```
%autosave 0
```

```
Autosave disabled
```

(A.1) Laden Sie NumPy, Pandas und Matplotlib.pyplot mit den üblichen Abkürzungen.

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

(A.2) Führen Sie den Magic Command aus, so dass Matplotlib-Plots "inline" erscheinen.

In [3]:
```
%matplotlib inline
```

Für die nächsten Aufgaben laden wir zuerst Daten zur Entwicklung einiger Teilindizes des Swiss Performance Index (SPI).

In [4]:
```
# Zeitreihen werden in Kapitel 11 erläutert.
Kurse = pd.read_csv('../weitere_Daten/hspitr.csv', sep=';',
                    usecols=[0, 2, 3, 4], parse_dates=[0],
                    index_col=0).sort_index()
# Von Tages- auf Jahresendkurse welcheln (siehe Kapitel 11):
Kurse = Kurse.resample('Y').last()
Kurse.head()
```

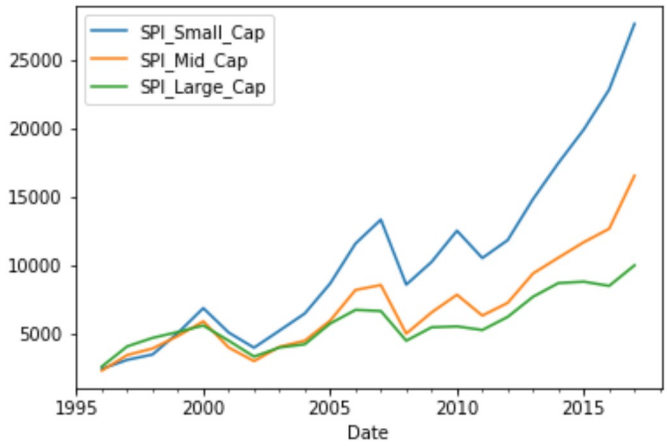
Out[4]:

Date	SPI_Small_Cap	SPI_Mid_Cap	SPI_Large_Cap
1996-12-31	2400.97	2277.34	2576.10
1997-12-31	3062.38	3414.00	4049.86
1998-12-31	3455.04	3895.44	4673.95
1999-12-31	5036.68	4812.13	5105.81
2000-12-31	6856.86	5886.65	5581.28

(B.1) Stellen Sie die drei Zeitreihen gemeinsam in einem Diagramm dar. Verzichten Sie vorerst auf jegliche Anpassungen des Defaults.

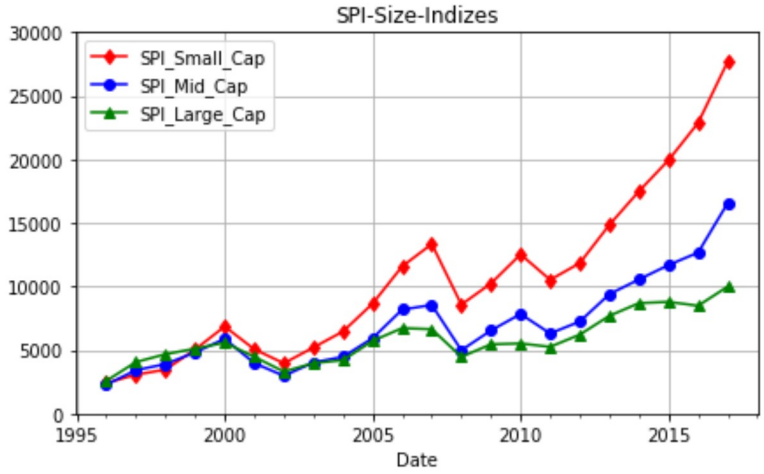
```
In [5]: Kurse.plot()
```

```
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x172272dd908>
```



(B.2) Stellen Sie sicher, dass Sie den Code in der folgenden Zeile verstehen. Sie können beispielsweise Argumente verändern und schauen, welche Wirkung sie haben.

```
In [6]: Kurse.plot(title='SPI-Size-Indizes',
                   figsize=(7, 4),
                   ylim=[0, 30000],
                   style=['rd-','bo-','g^-'])
        plt.grid()
        plt.legend(loc='upper left')
        plt.show()
```



(C.1) Laden Sie die Daten der Datei `drinksbycountry.csv` in das DataFrame `drinks` und lesen Sie die ersten 5 Zeilen aus. Die Datei befindet sich im Ordner "weitere_Daten".

In [7]:
```
drinks = pd.read_csv('../weitere_Daten/drinksbycountry.csv')
drinks.head()
```

Out[7]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	Asia
1	Albania	89	132	54	4.9	Europe
2	Algeria	25	0	14	0.7	Africa
3	Andorra	245	138	312	12.4	Europe
4	Angola	217	57	45	5.9	Africa

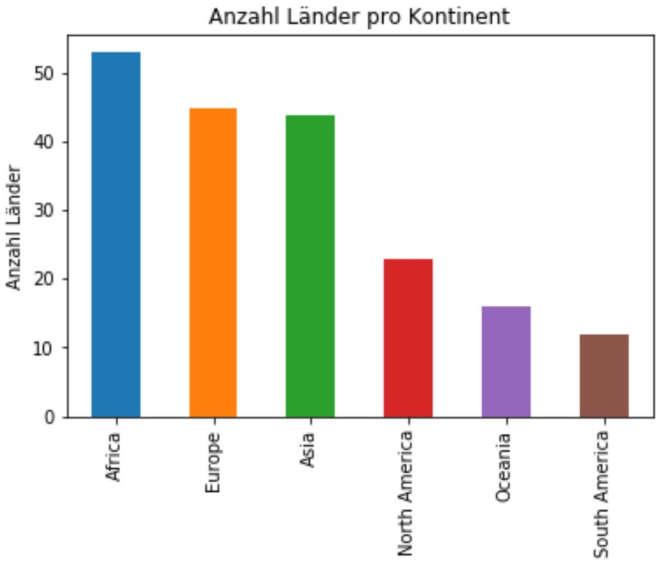
(C.2) Sellen Sie die Verteilung der Anzahl Länder pro Kontinent in einer Häufigkeitstabelle dar.

In [8]:
```
tab = drinks.continent.value_counts()
tab
```

Out[8]:
```
Africa           53
Europe           45
Asia             44
North America    23
Oceania          16
South America    12
Name: continent, dtype: int64
```

(C.3) Sellen Sie die Verteilung der Anzahl Länder pro Kontinent in einem Säulendiagramm dar. Geben Sie dem Diagramm einen geeigneten Titel und beschriften Sie die Ordinate passend.

In [9]:
```
tab.plot.bar(title='Anzahl Länder pro Kontinent')
plt.ylabel('Anzahl Länder')
plt.show()
```



Man kann mit Säulendiagrammen nicht nur Häufigkeitsverteilungen darstellen. Es ist beispielsweise auch möglich, Gruppendurchschnitte darzustellen. Betrachten wir hierzu den Datensatz `Auto.csv`, den wir zuvor kennengelernt haben.

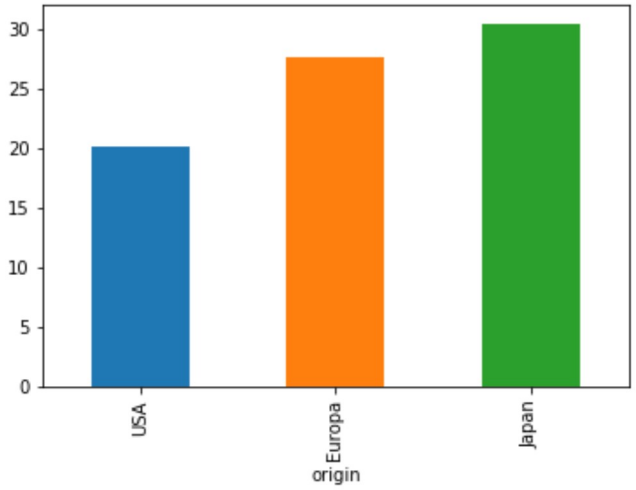
Die erste Befehlszeile liest die Daten ein. Gebraucht werden die Variablen `mpg` (Meilen pro Gallone) und `origin` (Herkunftsland der Autos), wobei diese als Index festgelegt wird. Danach werden die mittleren Meilen pro Gallone nach Herkunftsland berechnet und der Index umbenannt. (Im nächsten Kapitel erfahren wir, wie man mit der `groupby`-Methode eleganter hätte vorgehen können).

In [10]:
```
auto = pd.read_csv('../weitere_Daten/Auto.csv', sep=';',
                   usecols=[0,7], index_col=[1])
means = auto.mpg.mean(level=0).sort_index()
means.rename({1: 'USA', 2: 'Europa', 3: 'Japan'}, inplace=True)
means
```

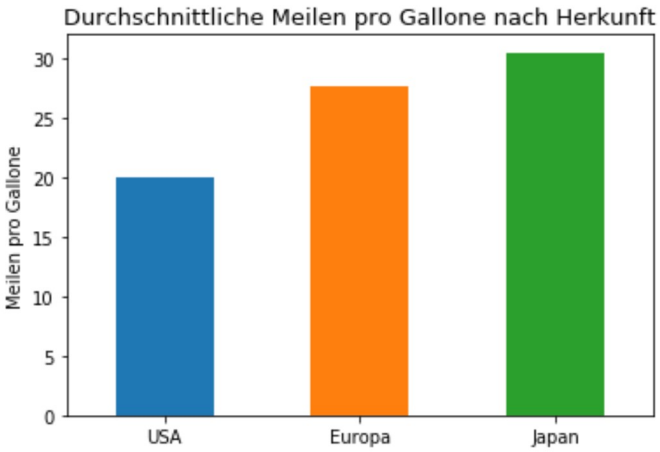
Out[10]:
```
origin
USA       20.033469
Europa    27.602941
Japan     30.450633
Name: mpg, dtype: float64
```

(C.4) Stellen Sie die gebildeten Mittelwerte mit einem passenden Diagramm dar.

In [11]:
```
means.plot.bar();
```

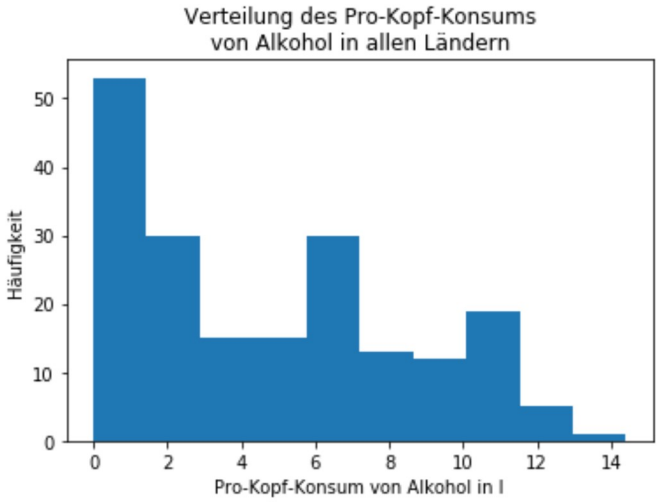


In [12]:
```
# Empfohlende Verbesserungen des Diagramms:
means.plot.bar(rot=0)  # x-Achsen-Ticks nicht rotieren
plt.title('Durchschnittliche Meilen pro Gallone nach Herkunft',
          fontsize=13)  # Grösse des Titels
plt.xlabel('')  # Keine x-Achsenbeschriftung
plt.ylabel('Meilen pro Gallone')
plt.show()
```



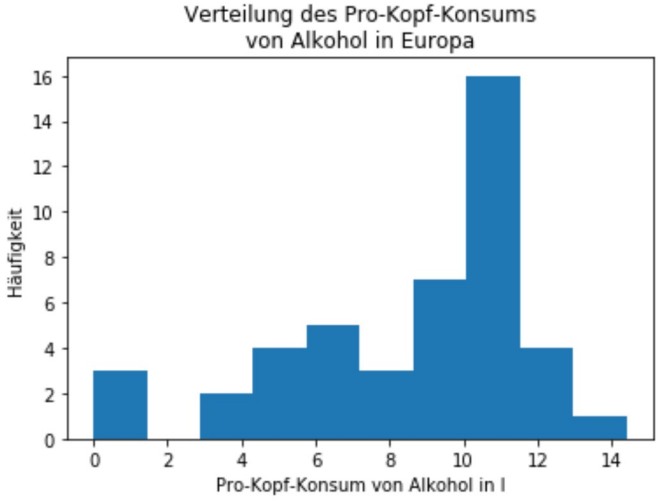
(D.1) Erstellen Sie ein Histogramm der Variable `total_litres_of_pure_alcohol` aus dem DataFrame `drinks`. Achten Sie auf korrekte Achsenbeschriftungen.

In [13]:
```
drinks.total_litres_of_pure_alcohol.plot.hist(
    title='Verteilung des Pro-Kopf-Konsums\nvon Alkohol in allen Ländern')
plt.xlabel('Pro-Kopf-Konsum von Alkohol in l')
plt.ylabel('Häufigkeit');
```



(D.2) Erstellen Sie nun das Histogramm nur für europäische Länder.

In [14]:
```
drinksEurope = drinks[drinks.continent=='Europe']
drinksEurope.total_litres_of_pure_alcohol.plot.hist(
    title='Verteilung des Pro-Kopf-Konsums\nvon Alkohol in Europa')
plt.xlabel('Pro-Kopf-Konsum von Alkohol in l')
plt.ylabel('Häufigkeit')
plt.show()
```



(D.3) Bestimmen Sie die drei Länder aus Europa, die den tiefsten Pro-Kopf-Konsum an Alkhol haben (im Histogramm ganz links). Was fällt Ihnen auf?

In [15]:
```
drinksEurope.sort_values('total_litres_of_pure_alcohol')[:3]
```

Out[15]:

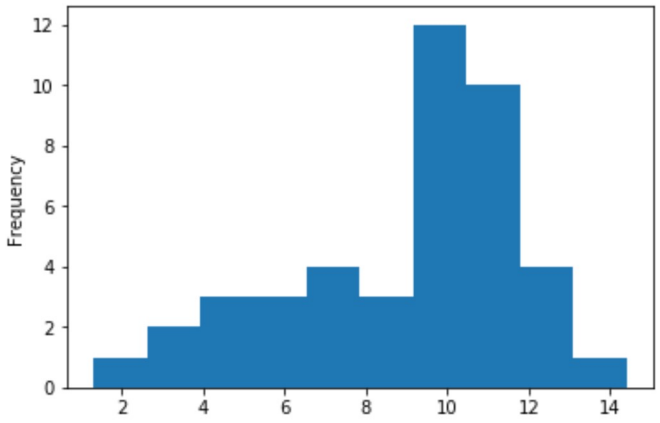
	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
147	San Marino	0	0	0	0.0	Europe
111	Monaco	0	0	0	0.0	Europe
10	Azerbaijan	21	46	5	1.3	Europe

Die Daten für San Marino und Monaco scheinen fehlerhaft zu sein (bzw. die Grössen wurden wohl nicht erfasst). Man sollte diese Länder aus der Untersuchung ausschliessen.

(D.4) Erstellen Sie nochmals das Histgogramm für Europa. Schliessen Sie aber Länder aus, wo der totale Alkoholkonsum nicht positiv ist. Das "Default-Histogramm" (ohne Anpassungen) reicht hier.

In [16]:
```
drinksEurope2 = drinksEurope[drinksEurope.total_litres_of_pure_alcohol>0]
drinksEurope2.total_litres_of_pure_alcohol.plot.hist()
```

Out[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x17227ef1e80>
```



(E.1) Laden Sie die Daten `Auto.csv` (Ordner "weitere_Daten") in das DataFrame `Auto` .

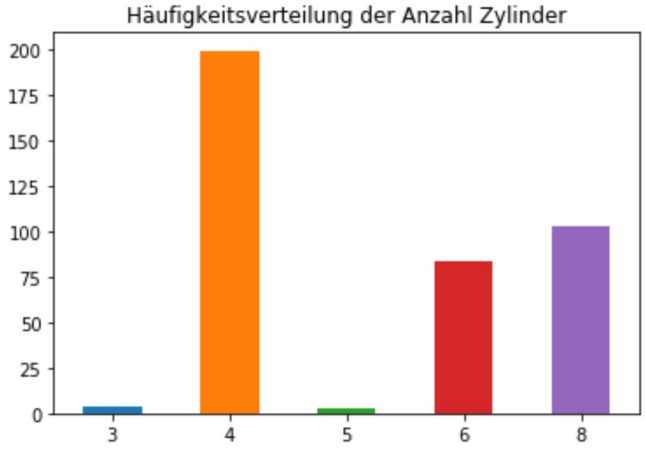
In [17]:
```
Auto = pd.read_csv('../weitere_Daten/Auto.csv', sep=';')
Auto.head()
```

Out[17]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino

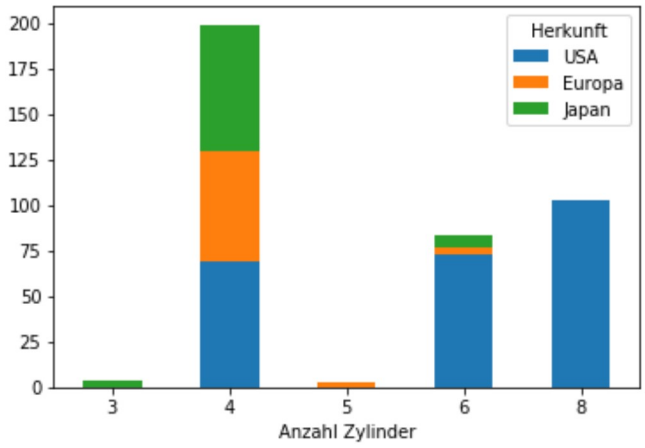
(E.2) Stellen Sie die Verteilung der Anzahl Zylinder in einem Säulendiagramm dar. Achten Sie darauf, dass die Zylinder auf der x-Achse sortiert sind (mit `sort.index()`).

In [18]:
```
Auto.cylinders.value_counts().sort_index().plot.bar(
            title='Häufigkeitsverteilung der Anzahl Zylinder',
            rot=0);
```



(E.3) Betrachten Sie die folgende Abbildung (also nur das Ergebnis der Zelle). Interpretieren Sie.

In [19]:
```
tab = Auto.groupby('origin').cylinders.value_counts().unstack().T
tab.rename(columns={1: 'USA', 2: 'Europa', 3: 'Japan'}, inplace=True)
tab.columns.name = 'Herkunft'
tab.index.name = 'Anzahl Zylinder'
tab.plot.bar(rot=0, stacked=True);
```



Antworten:

- Es werden vorwiegend Autos mit gerader Anzahl Zylinder gebaut (was verständlich ist).
- Am häufigsten sind Autos mit 4 Zylindern.
- Autos aus Europa und Japan haben fast ausschliesslich 4 Zylinder (im Datensatz!)
- Mehr als 4 Zylinder findet man fast ausschlisslich in US-Autos, dort aber sehr häufig. 8 Zylinder ist sogar der Modus bei US-Autos! (Zu berücksichtigen ist, dass die Autos aus den 1970er und 1980er Jahren sind!)

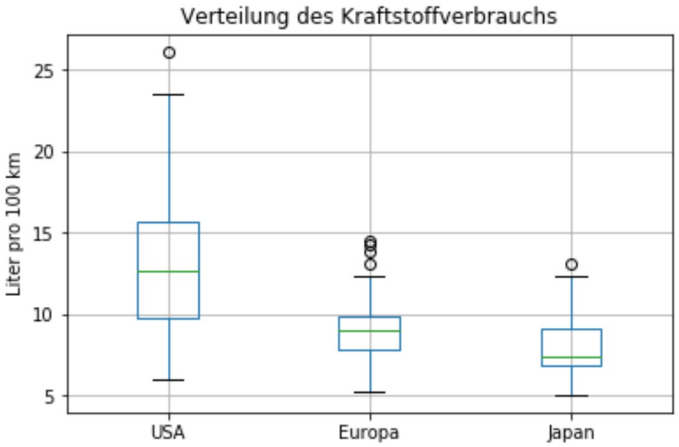
(E.4) Erstellen Sie im DataFrame `Auto` aus der Variable `mpg` (Miles per Gallon) die Variable `Verbrauch` (*Liter pro 100 km*). (Hinweis: 1 Meile = 1.60934 km, 1 Gallone = 3.78541 l)

In [20]:
```
Auto['Verbrauch'] = (100*3.78541/1.60934)/Auto.mpg
Auto.Verbrauch.describe()
```

Out[20]:
```
count    392.000000
mean      11.248555
std        3.913846
min        5.047533
25%        8.110864
50%       10.340372
75%       13.836180
max       26.135006
Name: Verbrauch, dtype: float64
```

(E.5) Vergleichen Sie den Kraftstoffverbrauch der Autos nach Herkunft (`origin` ; 1 für USA, 2 für Europa und 3 für Japan). Erstellen Sie hierzu einen gruppierten (faktorisierten) Boxplot. Beschriften Sie das Diagramm sinnvoll.

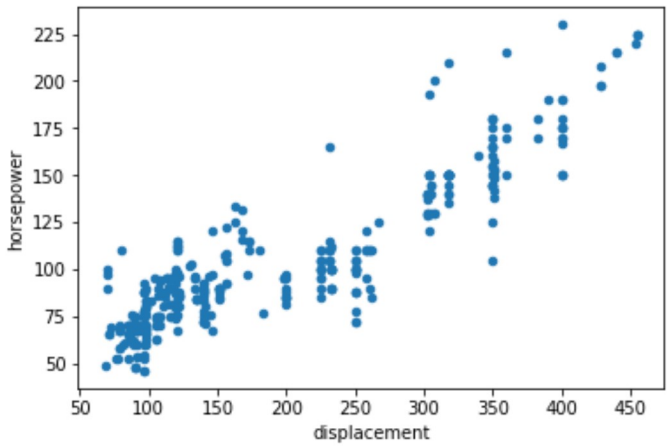
In [21]:
```
Auto.boxplot(column='Verbrauch', by='origin')
plt.title('Verteilung des Kraftstoffverbrauchs')
plt.xlabel('')
plt.xticks([1,2,3], ['USA', 'Europa', 'Japan'])
plt.ylabel('Liter pro 100 km')
plt.suptitle('');
```



Der Verbrauch (Liter pro 100 km) von US-Autos ist deutlich höher als von Autos aus Europa und Japan.

(E.6) Untersuchen Sie die Beziehung zwischen Hubraum (`displacement`) und PS (`horsepower`) mit einem geeigneten Diagramm **und** einer geeigneten Kennzahl.

In [22]:
```
Auto.plot.scatter('displacement', 'horsepower');
```



In [23]:
```
Auto.displacement.corr(Auto.horsepower).round(3)
```

Out[23]: 0.897

Es gibt einen starken positiven linearen Zusammenhang zwischen Hubraum und PS.

Ende der Übung